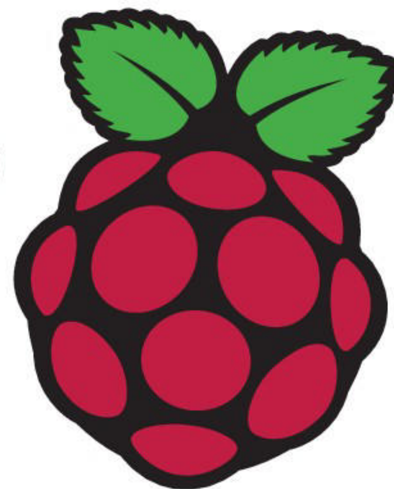


VISITA WWW.RASPBERRYITALY.COM

The MagPi



Numero 80 | Aprile 2019 | magpi.cc raspberryitaly.com

La rivista ufficiale Raspberry Pi tradotta in italiano da RaspberryItaly

I 50 BEST RASPBERRY PI TIPS

HACK ESSENZIALI, SUGGERIMENTI E TRUCCHI

HAM RADIO

Raspberry Pi e il mondo dei Radioamatori

PROGRAMMA UN GIOCO ISOMETRICO

Terza e ultima parte del tutorial sulla creazione del gioco AmazeBalls



The MagPi magazine is published by Raspberry Pi (Trading) Ltd., Mount Pleasant House, Cambridge, CB3 0RN. ISSN: 2051-9982.

I 50 BEST RASPBERRY PI TIPS

I 50 MIGLIORI TRUCCHI RASPBERRY PI

PADRONEGGIA RASPBIAN E RASPBERRY PI CON QUESTI HACKS, SUGGERIMENTI E TRUCCHI

01 Boot da USB

Invece di unsare una scheda SD, puoi avviare alcuni modelli Pi (3B, 3B+, 3A+, 2B v1.2) da una porta USB. Sarà più veloce e più affidabile per un uso a lungo termine, in più, otterrai molto spazio di archiviazione aggiuntivo. Vedi la documentazione ufficiale su magpi.cc/eXHfjq.

02 Power over Ethernet

L'HAT ufficiale Raspberry Pi PoE HAT ti consente di alimentare il Raspberry Pi 3 B+ con un cavo Ethernet. Ottimo per quei progetti che richiedono una costante connessione cablata.

03 Uscita Audio

Tutti i modelli Pi hanno una uscita video composita per la connessione a dispositivi analogici. i Pi B+ e successivi combinano l'uscita audio e quella composita nella stessa presa jack da 3,5 mm. Questo richiede un particolare tipo di jack, con il canale audio sinistro sulla punta, audio destro sull'anello 1, la massa sull'anello 2 e il video sul manicotto. vedi magpi.cc/tDfjLV.

04 Cambiare opzioni audio

La porta HDMI emette anche audio, oltre al video. Se il tuo display non incorpora un altoparlante, vorrai usare il jack analogico per l'audio. In tal caso, fai clic con il pulsante destro del mouse sull'icona del volume nella barra dei menu e passa da HDMI ad analogico. In alternativa, usa la riga di comando:

```
amixer cset numid=3 1
```

Imposta l'ultimo numero su "1" per analogico, o "2" per HDMI. Usa raspi-config per cambiare l'impostazione audio all'avvio:

```
sudo raspi-config
```

Scegli Advanced Options > Audio e 'Force 3.5 mm ('headphone') jack' o 'Force HDMI' per impostarlo come predefinito.

05 Alimentarlo

I requisiti di alimentazione di Raspberry Pi aumentano a seconda di come fai uso delle varie interfacce. I pin GPIO, in totale, possono erogare in modo sicuro 50 mA, distribuiti su tutti i pin; un pin GPIO singolo può erogare solo 16 mA. La porta HDMI richiede 50 mA, il modulo videocamera 250 mA, tastiera e mouse possono volere solo 100 mA o poco più. Acquista l'alimentatore ufficiale e controlla la potenza dei tuoi dispositivi: **magpi.cc/xfszUR**.

06 Usare la camera

Puoi usare una webcam, ma Raspberry Pi ha una opzione dedicata: il modulo videocamera (**magpi.cc/jbKzbf**). Si collega alla CSI (Camera Serial Interface). Il modulo videocamera è usato per catturare video ad alta definizione e anche immagini. È facile da usare per i principianti, ma ha molto da offrire agli utenti avanzati, se stai cercando di espandere la tua conoscenza. Ci sono molti esempi online di persone che lo usano per video time-lapse, slow motion e altri usi intelligenti. Puoi anche usare le librerie che Raspberry Pi ha sviluppato per creare effetti visivi.

07 Touchscreen

Il display touch Raspberry Pi (**magpi.cc/CAPxtI**) offre agli utenti la possibilità di creare progetti integrati tutto-in-uno come tablet, sistemi di infotainment e progetti embedded. Il suo cavo a nastro si connette alla porta DSI presente su tutte le schede Raspberry Pi (tranne Raspberry Pi Zero e Zero W).

08 Guida ai pin GPIO

Apri un Terminale e digita:

```
pinout
```

...per una veloce guida visuale ai piedini del GPIO. oppure digita:

```
pinout -x
```

...per aprire il browser web sul modello interattivo su pinout.xyz.

09 Schemi Hardware

Puoi vedere i disegni meccanici del Raspberry Pi 3B+ (e di tutti gli altri modelli) sul sito Web Raspberry Pi: **magpi.cc/WWFUji**.

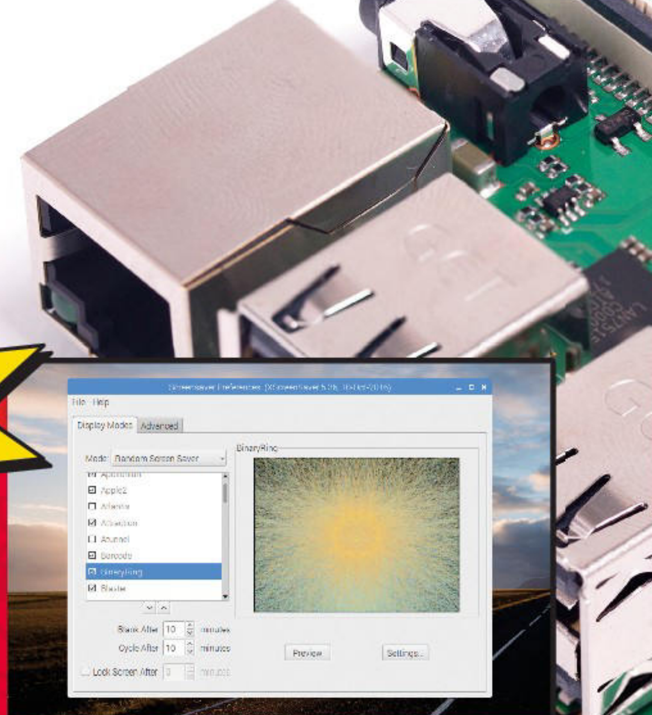
10 Hotspot wireless

Molti modelli di Raspberry Pi dispongono di un adattatore LAN wireless per connettersi alla rete locale. Ma puoi anche usare l'access point per creare un hotspot wireless. Il processo è un po' lungo, ma è fantastico per imparare di più su come funziona la LAN wireless. Leggine di più su **magpi.cc/nnriRT**.



TRUCCHI RASPBIAN

Personalizza il sistema operativo di Raspberry Pi con questi suggerimenti Trucchi. **Di Lucy Hattersley**



Aggiungi qualche tocco visivo all'interfaccia di Raspbian usando il programma salvaschermo

11 Programmi demo

Raspbian ha un sacco di programmi demo che dimostrano varie caratteristiche. Trovali qui:

```
cd /opt/vc/src/hello_pi
```

Sono scritti in C. Compilali con:

```
./rrebuild.sh
```

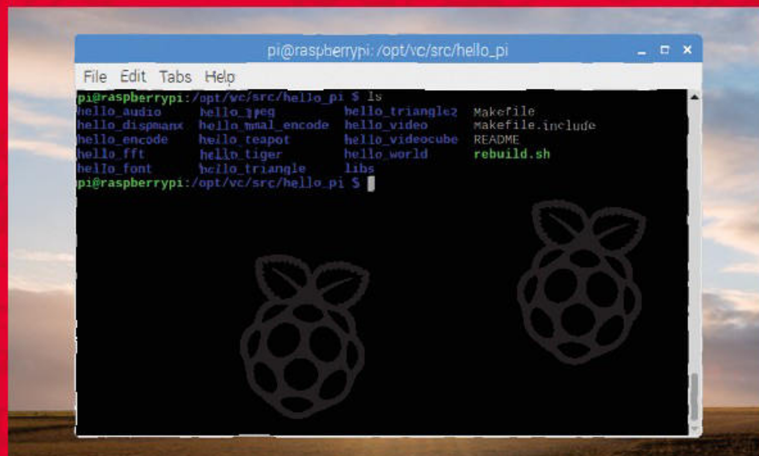
Ora sarai in grado di eseguire i file .bin in ogni cartella. Ad esempio, esegui:

```
cd hello_tiger
./hhello_tiger.bin
```

... per vedere sullo schermo una grande immagine rotante di una tigre. Per maggiori informazioni, vedi **magpi.cc/tpNaGV**.

12 Screensaver

Di default, il desktop Raspberry Pi non ha un software per screensaver facile da usare installato. In primo luogo, dovresti installare l'applicazione screensaver di X Windows.




```
sudo apt-get update
sudo apt-get install xscreensaver
```

Potrebbe richiedere alcuni minuti. Poi apparirà l'applicazione screensaver elencata nel menu principale del desktop. Questa fornisce molte opzioni per configurare lo screensaver o disabilitarlo.

13 Capire config.txt

Nota: questo è solo per utenti esperti! Il Pi usa un file di configurazione invece del BIOS che trovi su un PC convenzionale. Questo file è normalmente accessibile come /boot/config.txt da Linux, e deve essere modificato come root.

```
sudo nano /boot/config.txt
```

Qualsiasi modifica avrà effetto solo dopo che avrai riavviato il Raspberry Pi. Dopo che Linux è partito, puoi visualizzare la impostazioni correnti usando i seguenti comandi:

```
vcgencmd get_config <config>
```

...mostra un valore di configurazione specifico; esempio, `vcgencmd get_config arm_freq`.

```
vcgencmd get_config int
```

...elenca tutte le opzioni di configurazione impostate con un intero (diverse da zero).

```
vcgencmd get_config str
```

... elenca tutte le opzioni di configurazione delle stringhe impostate (non nulle).

Un sacco di opzioni possono essere modificate nel file config.txt – vedi rpf.io/configtxt.

TRUCCHI SICUREZZA

Quando connetti il tuo Pi al mondo esterno, devi tenere conto della sicurezza

14 Libera spazio

Puoi risparmiare spazio sulla tua scheda SD per progetti, documenti e programmi, rimuovendo alcuni dei programmi forniti di default in Raspbian. Puoi rimuovere il software dalla riga di comando:

```
sudo apt purge mathematica
sudo apt clean
sudo apt autoremove
```

Ma è molto meglio scegliere Menu> Preferenze> Software consigliato. Togli il segno di spunta accanto al programma indesiderato e fai click su OK.

15 Aggiornamento kernel

Se utilizzi il processo standard di update / upgrade di Raspbian, il kernel sarà aggiornato automaticamente all'ultima versione stabile. È la procedura consigliata. Però, a volte, potresti voler aggiornare all'ultimo 'bleeding edge' o kernel di test.

```
sudo rpi-update
```

L'utility rpi-update scaricherà l'ultima versione del file del kernel (instabile, testata) e copierà tutti i file richiesti sul tuo sistema. Nota che non è garantito il corretto funzionamento dell'ultimo kernel di rpi-update! Assicurati che non vada in conflitto con il tuo pacchetto di distribuzione. Non fornisce un sistema per disinstallare automaticamente il file.

Per ripristinare l'attuale kernel stock di Raspbian, dopo aver provato rpi-update o un Kernel custom, è necessario reinstallare entrambi questi pacchetti, eseguendo:

```
sudo apt-get install --reinstall
raspberrypi-bootloader raspberrypi-
kernel
```

16 Cambia la password

Con l'installazione standard di Raspbian, il nome utente predefinito è **pi** e la password è **raspberry**. Se non hai modificato almeno questa password, chiunque può accedere al tuo Pi! Vai in Menu> Preferenze> Configurazione Raspberry Pi. Nella scheda Sistema, fai clic su Cambia Password, immettine una, confermala e fai clic su OK.

17 Tieni Raspbian aggiornato

Di tanto in tanto, vulnerabilità di sicurezza vengono trovate nei software, quindi è sempre meglio Aggiornare regolarmente alle ultime versioni. Nel Terminale, digitare `sudo apt-get update` per aggiornare gli elenchi dei pacchetti, quindi `sudo apt-get dist-upgrade` per avere sul tuo sistema l'ultima versione dei pacchetti.

18 Installa Fail2ban

Se qualcuno vuole hackerare il tuo Pi, potrebbero provare a indovinare il tuo nome utente e la password. Probabilmente ci vorranno molti tentativi, questo attacco è chiamato 'brute-force'. Per prevenirlo, è possibile installare un programma chiamato Fail2ban. Installalo usando `sudo apt install fail2ban` e un utente sarà bannato per dieci minuti se fallisce il login cinque volte.

19 Cambia la porta di SSH

Se cambi la porta SSH predefinita, chiunque provasse a connettersi, dovrebbe sapere quale porta usare. Per modificarla, `sudo nano /etc/ssh/sshd_config` e modifica la riga #Port 22 in Port 2222 (senza #) o un altro numero che ricordi. Salva ed esci (**CTRL+O, INVIO, CTRL+X**), quindi riavvia SSH con `sudo service ssh restart`.

20 Spegni le interfacce che non ti servono

Il modo più sicuro per fermare gli hacker è non avere nessuna connessione di rete, ma un altro modo di disabilitare un po' il sistema è quello di andare in Menu principale ->> Preferenze ->> Configurazione Raspberry Pi e selezionare la scheda Interfacce. Assicurati che tutto ciò che non ti serve sia disabilitato.

SUGGERIMENTI PROGRAMMAZIONE E MAKING

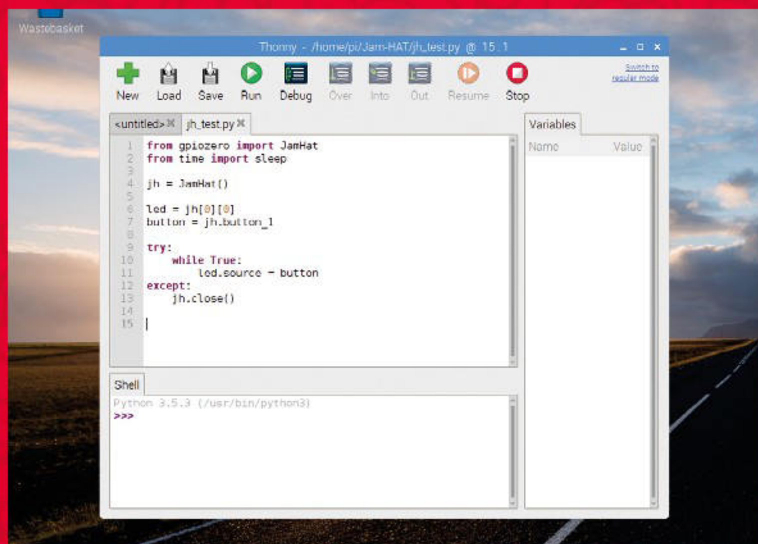
Ottieni di più dalle tue creazioni con Raspberry Pi. **Di Lucy Hattersley**

21 Passa a Thonny

Se stai ancora programmando con IDLE Python o usando un editor di testo, passa a Thonny. È facile come gli altri programmi, ma ha un semplice debugger che rende molto più facile correggere i difetti nel tuo codice. Premi **CTRL+F5** invece di **F5** per eseguire un programma passo-passo. A differenza degli IDE più avanzati, non è necessario impostare punti di interruzione. Premi F6 per fare un grande passo e F7 per un piccolo passo. È ideale per i maker perché ti consente di risolvere i problemi, ma non è così impegnativo da essere completamente assorbito dal codice e così puoi continuare con la tua realizzazione.

22 GPIO Zero

Assicurati di imparare a programmare usando la libreria Python GPIO Zero (ora in versione 1.5). Questa libreria è stata progettata per togliere ogni seccatura nell'uso dei kit e componenti elettronici con Raspberry Pi. Con GPIO Zero puoi passare più tempo a mettere insieme il tuo progetto e meno tempo a capire come farlo funzionare. Metti la documentazione tra i preferiti: **gpiozero.readthedocs.io**.



▲ L'IDE (Integrated Development Environment) Thonny rende più facile eseguire il debug dei tuoi programmi senza avere troppe distrazioni

23 Fare schemi circuitali

Ti piacerebbe fare uno schema circuitale come quelli che vedi sulla rivista *The MagPi*? Il software che stai cercando è Fritzing. È attualmente in beta e per lo più noi utilizziamo la versione Windows per creare le immagini che vedi nella rivista. Tuttavia, è disponibile una versione per Raspberry Pi.

```
sudo apt-get update
sudo apt-get install fritzing
```

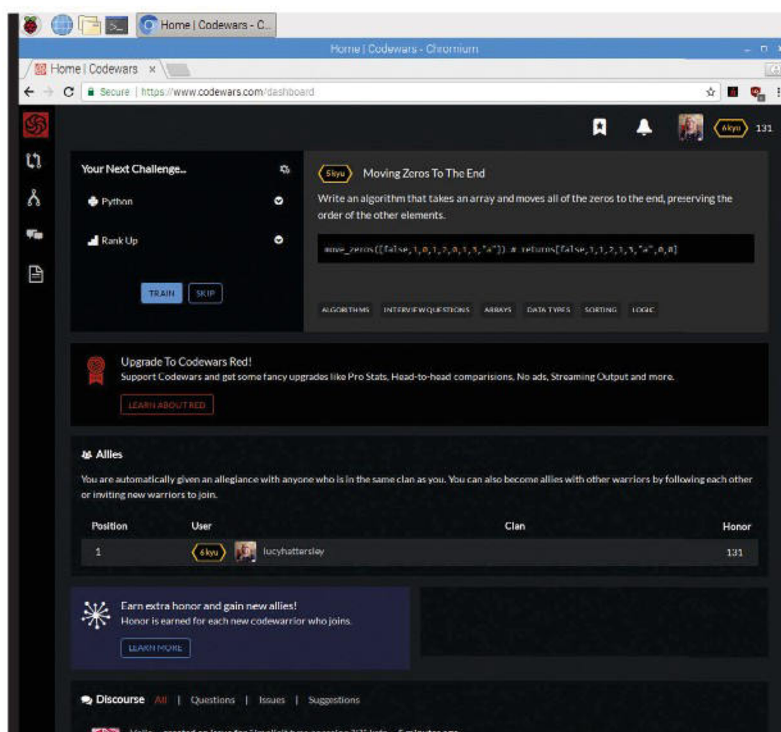
Riavvia il tuo Raspberry Pi e troverai il programma in Menu> Programmazione. Attualmente stiamo avendo problemi ad aprire le parti bin sul Raspberry Pi; se qualche lettore ha una soluzione, per favore fateci sapere.

24 Raspberry Pi e Arduino

Raspberry Pi funziona straordinariamente bene collegato a un Arduino. Nonostante siano visivamente simili, le due schede sono molto diverse. Raspberry Pi è un computer completo su singola scheda, mentre la gamma Arduino è composta da microcontrollori. Funzionano perfettamente bene insieme perché Arduino è utile per il controllo dei componenti, e Raspberry Pi è migliore per le interfacce di livello superiore e controllo principale. Puoi installare il software per controllare un Arduino da Raspberry Pi:

```
sudo apt-get upgrade
sudo apt-get install arduino
```

I programmi Arduino sono chiamati 'sketches' e sono basati sul linguaggio di programmazione C. Per maggiori info, vedi il nostro libro sull'apprendimento C (**magpi.cc/learn-c-book**) e la nostra guida a Arduino e Pi sul numero 67 (**magpi.cc/67**).




25 Codewars

Uno dei modi migliori per migliorare le tue competenze di programmazione è quello di completare delle sfide. Se stai cercando delle sfide di programmazione, iscriviti a Codewars sul tuo Raspberry Pi. Il sito web (**codewars.com**) ha lo stile di un videogioco: ottieni punti e avanzerai nei livelli completando le sfide (kata). Abbiamo persino visto usarlo in colloqui di lavoro e per ottenere l'accesso ai boot camp di programmazione.

26 Impara a saldare

C'è una tecnica per saldare i fili tra loro chiamata "Western Union" o giuntura 'Lineman' e dovresti impararla se poi salderai (**magpi.cc/mckKVW**). È stata sviluppata con l'introduzione del telegrafo (da cui il nome) e il tipo di avvolgimento è progettato per stringersi quando le due estremità vengono tirate.

27 Realizza un kit per maker

Non possiamo non consigliare vivamente di mettere assieme un kit per maker. Questo dovrebbe contenere una vasta gamma di parti elettroniche, minuterie e strumenti. Trova un buon contenitore per componenti e inserisci una selezione di resistenze, LED, condensatori, pulsanti, potenziometri, un cicalino piezoelettrico e una breadboard con un sacco di cavi jumper. Puoi trovare kit e parti nella maggior parte dei rivenditori Raspberry Pi o visitare il negozio Raspberry Pi a Cambridge. Noi usiamo una breadboard a mezza grandezza nei nostri schemi, quindi se trovi quel tipo, troverai molto più facile seguire i nostri progetti: **magpi.cc/dXHByV**.

28 Seconda mano

Frequenta il negozio/mercatino di seconda mano della tua zona. Gli oggetti usati possono essere una grande fonte di ispirazione, e fare upcycling della vecchia tecnologia con nuove parti Raspberry Pi è un passatempo popolare. Il maker di alto profilo Martin Mander ha salvato vecchie radio, televisori e giocattoli. Seguilo su Twitter (**@MartinWMander**) per trarre ispirazione dalle sue idee.

29 Condividi presto e spesso

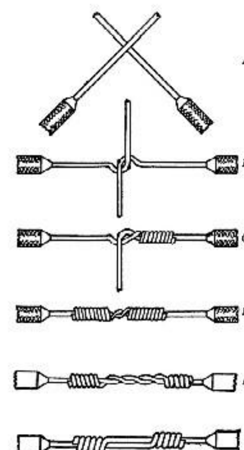
Il modo migliore per ottenere ispirazione e aiuto per il tuo progetto è di condividerlo con la Comunità Raspberry Pi. con l'ultimo numero abbiamo lanciato #MonthOfMaking su Twitter, per incoraggiare i nostri lettori a condividere progetti tra loro. Ma non aspettare fino al prossimo anno - condividi il tuo progetto sul nostro forum (forum.raspberryitaly.com), Facebook (facebook.com/raspberryitaly) o Twitter (@TheMagPi). Più altri lettori vedono il tuo progetto, più probabilmente avrai opportunità di finirlo.

30 Suggerimenti per maker

Element14 ha due raccolte di suggerimenti per i maker che sono pieni di idee interessanti. Iscriviti al sito Web di Element14 per accedervi: **magpi.cc/XQaCyC**.

▲ Codewars è una buona risorsa web per allenare i tuoi muscoli programmatori

▼ Imparare la tecnica della giunzione 'Western Union' renderà ancora più forti i tuoi giunti di saldatura





TRUCCHI DI RETE

Vuoi saperne di più sulle connessioni di rete di Raspberry Pi? Ecco alcune basi e qualche suggerimento interessante se sei già connesso in rete

Puoi connettere il tuo Pi a una LAN e a Internet utilizzando un cavo RJ45 o configurando il WiFi

Cosa Serve

- Cavo di rete
- Router internet
- Connessione internet

31 Cablato o wireless

Il Raspberry Pi ha due metodi standard per connettersi a una rete: con un cavo Ethernet o con il WiFi integrato (sui modelli più recenti). Il cavo Ethernet è in genere di Cat5e o Cat6 (per reti veloci), con un connettore RJ45 su ciascuna estremità. Una collegata a un router o hub e l'altra estremità al Raspberry Pi. Le connessioni cablate sono generalmente più veloci del WiFi e si connetterà, se possibile, non appena viene collegato il cavo. la connessione alla rete wireless domestica può generalmente essere effettuata selezionando la rete e digitando la relativa password - vedi **magpi.cc/HiDkyk**.

32 Scopri il tuo IP

Un indirizzo IP (Internet Protocol) è un numero che viene assegnato al tuo Raspberry Pi in modo che altri computer sappiano dove trovarlo sulla rete. Ci sono diversi modi per trovare l'indirizzo IP corrente del tuo Pi. In una finestra del Terminale, puoi usare un paio di comandi. il vecchio comando (che funziona ancora) è `ifconfig`. Ora c'è un nuovo comando più breve, `ip`, che ritornerà delle informazioni simili. Il tuo indirizzo IP locale sarà un numero tipo 192.168.0.34. Puoi anche trovare il tuo indirizzo IP passando con il mouse sopra l'icona di rete (in alto a destra) del desktop.

Puoi usare il strumento curl per scaricare i file, come i PDF dei numeri di *The MagPi*

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ curl -O https://www.raspberrypi.org/magpi-issues/MagPi79.pdf
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
 41 31.8M   41 13.1M    0     0  3072k      0  0:00:10  0:00:04  0:00:06 3072k
```

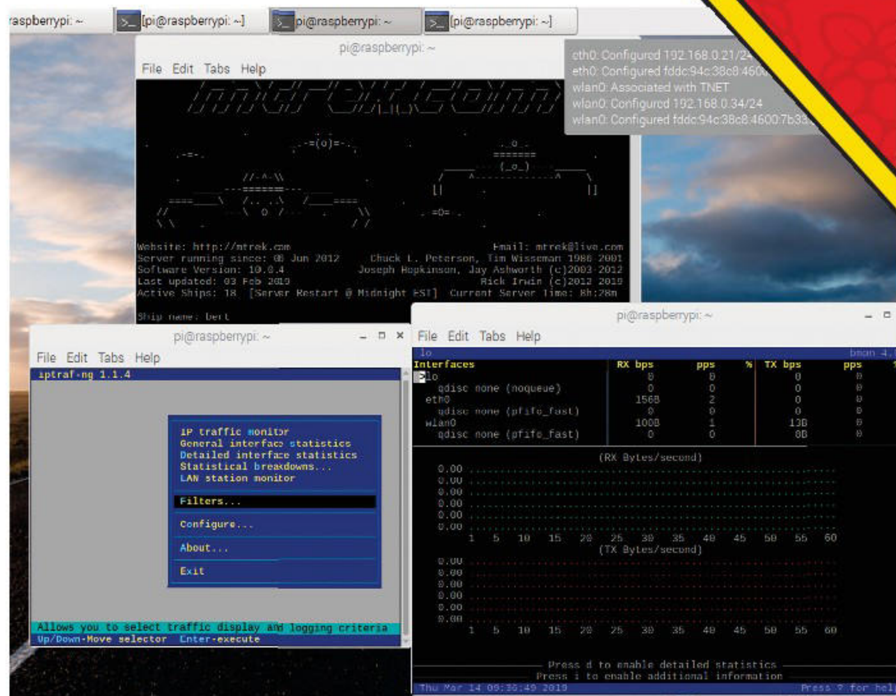
33 IPv4 e IPv6

I termini IPv4 e IPv6 sono descrizioni del protocollo internet e dell'indirizzo. IPv4 utilizza numeri a 32 bit per l'indirizzo e assomiglia a: 192.168.1.23. Siccome Internet è diventato così grande, il numero limitato di indirizzi IPv4 disponibili è diventato un problema. È stata quindi definita una nuova versione di IP (IPv6), che utilizza 128 bit per l'indirizzo IP. A luglio 2017, è stata pubblicata la versione finale del protocollo, sebbene IPv6 sia in uso da oltre 20 anni. Un indirizzo IPv6 assomiglia a: fe80: 94c: 38c8: 4600: 7b33: 64bb: baf7: e866. In una rete domestica utilizzerai normalmente Indirizzi in formato IPv4, ma fai attenzione che in futuro, questo potrebbe cambiare.

34 Comandi utili

Puoi fare parecchio con i comandi di rete nel Terminale. Alcuni comandi devono essere prima installati. Il comando ping è invece presente di default. Se scrivi `ping` e poi un indirizzo IP o un dominio web, ti dirà quanto tempo ci vuole per inviare un messaggio a quel computer/server e quindi a ottenere una risposta. Anche Cat è preinstallato. Scrivi `cat /sys/class/net/wlan0/operstate` per vedere lo stato della tua connessione WiFi. Installa whois con `sudo apt-get install whois` e poi digita `whois` con un nome di dominio e troverai i dettagli di registrazione di quel dominio.





35 File di rete

Raspberry Pi ha diversi file in cui sono contenuti i dettagli sulla rete. È abbastanza utile sapere dove sono alcuni di questi, perché potrebbe essere necessario modificare alcune informazioni per delle configurazioni personalizzate. Il file **/etc /hosts** ha una lista di nomi che mappano gli indirizzi IP. Potresti avere una voce '192.168. 0.3 myserver' e questo significa che se digiti "myserver" nella barra degli indirizzi di un browser webr, la rete prova a connettersi a 192.168.0.3. Altri file utili sono **/etc/networks**, che ha i nomi di rete mappati con indirizzi IP; **/etc/protocols**, che ha una lista di tutti i protocolli di rete che possono essere usati; e **/etc/services**, che ha una lista di tutti i servizi di rete disponibili.

36 Traffico di rete

Se vuoi sapere qualcosa su quel che accade tra il tuo Pi e il resto della rete, ci sono alcuni comandi che puoi installare. Come un semplice comando statistico con `sudo apt-get install vnstat`. Quando lanci questo comando, otterrai alcuni dati di base di trasmissione totali. Se vuoi alcuni strumenti extra, è possibile installare **iptraf-ng**, che ha un menu con diversi strumenti e filtri per analizzare il traffico di rete. Per vedere alcune statistiche di trasferimento dati in tempo reale, installa **bmon**, che mostra le quantità di dati al secondo.

37 Telnet

Sebbene Telnet sia un sistema per le reti molto vecchio (fu progettato per la prima volta nel 1969), è ancora disponibile e può essere installato su Raspberry Pi utilizzando `sudo apt-get install telnet`. Telnet è un po' come una versione molto antica dei siti Web e quando è installato, il tuo Pi può connettersi a qualsiasi server Telnet. Per esempio, digita `telnet towel.blinkenlights.nl 23` per vedere una versione ASCII di Star Wars, o prova `telnet mtrek.com 1701` per un gioco di Star Trek. Ci sono ancora molti siti Telnet attivi là fuori, alcuni dei quali possono essere trovati su: **telnet.org/htm/places.htm**.

38 PiServer

PiServer è uno strumento utile per connettere tra loro molte schede Raspberry Pi senza bisogno di che vi siano inserite schede SD. Per informazioni su come configurarlo, vedi **magpi.cc/RQDQXx**.

39 Scaricare file con curl

Puoi scaricare i file direttamente dal Terminale con lo strumento curl. Usa `curl -O <URL>`. Vedi **magpi.cc/JrKoTA** per maggiori dettagli.

40 Headless wireless

Se usi il Raspberry Pi in modalità 'headless' (senza interfaccia Raspbian), può essere complicato connettersi a una rete wireless. Dovrai editare un file **wpa_supplicant.conf** con i dati della tua rete wireless e poi metti questo file nella cartella **boot**. Vedi qui per maggiori informazioni: **magpi.cc/RHviuV**.

▲ Puoi usare gruppi di diversi strumenti per analizzare le connessioni di rete del tuo Raspberry Pi; puoi persino giocare a giochi online nella finestra del Terminale!



TRUCCHI LINEA DI COMANDO

Stai cercando come giocare con la linea di comando? **PJ Evans** introduce i più popolari trucchi e scorciatoie

41 Piping

Le applicazioni nei sistemi Linux, come Raspbian, hanno tre modi di comunicare con te e con le altre applicazioni: standard in (stdin), standard out (stdout) e standard error (stderr). L'output di un'applicazione della riga di comando arriva sullo stdout e puoi usare il carattere pipe (tubo) '|' per inviare direttamente quell'output a un'altra applicazione, come questa:

```
ls -l | wc -l
```

Qui otteniamo un list della directory corrente in forma estesa e "intubiamo" questo output verso "wc", una applicazione di conteggio. L'argomento -l significa 'conta le linee', quindi ora abbiamo una 'nuova' app che ci dice quanti oggetti ci sono in una directory. Puoi concatenare tutti i comandi che desideri per realizzare compiti complessi.

42 Reindirizzare l'output

Prova questo:

```
curl https://cataas.com/cat
```

Il comando scarica l'immagine di un gatto – ma sei sul Terminale, che non può mostrare grafica. L'immagine arriva sotto forma di caratteri confusi in stdout. Possiamo usare il re indirizzamento per salvala come file, tipo questo:

```
curl https://cataas.com/cat > cat.jpg
```

L'operatore '>' dice a Raspbian di mettere tutto quello che arriva sullo stdout nel file invece che sul tuo schermo. Puoi anche usare '>>' che si aggiunge a un file esistente o '<' per dare in pasto a un comando il contenuto di un file.

43 Lanciare comandi multipli

Puoi inserire comandi multipli su una singola riga in modo che vengano eseguiti uno dopo l'altro:

```
command1; command2; command3
```

Per proseguire solo se il comando precedente è riuscito:

```
command1 && command2 && command3
```

Per esempio:

```
mkdir ~/myDir && cd ~/myDir && nano myFile.txt
```

... creerà una directory, vi entrerà e aprirà l'editor di testo nano con un nuovo file, myFile.txt. Puoi salvare questa linea da qualche parte per uso futuro o creare un alias (vedi suggerimento 47).

44 Lanciare task in background

L'esecuzione di un comando "in background", ti consente di lanciare, mentre funziona, altri comandi. Se stai usando il desktop, la soluzione più semplice è aprire un'altra finestra del Terminale. In alternativa, per eseguire un processo in background, aggiungi una e commerciale (&) alla fine del comando:

```
very-long-command &
```

Funziona bene, ma qualsiasi output su stdout e stderr apparirà sullo schermo della tua console. Per evitarlo, usa 'nohup':

```
nohup very-long-command &
```

Ora tutto l'output viene scritto nel file nohup.log nella tua directory attuale.

45 Guardare i file di log

A volte quando un processo sta generando un output, potrebbe essere necessario seguire ciò che sta accadendo. Ad esempio, il tuo server web non funziona perfettamente, così puoi trovare il problema guardando i log. Per mostrare il contenuto di qualsiasi file sullo schermo:

```
cat text.txt
```

Tuttavia, se il file di registro è enorme, avrai difficoltà a gestirlo. Per mostrare il file in "pagine":

```
less text.txt
```

Per vedere solo le ultime righe:

```
tail text.txt
```

Meglio ancora, per "seguire" il file di registro man mano che cresce:

```
tail -f text.txt
```

46 Scorciatoie miste

Tilde – Il carattere tilde '~' si riferisce sempre al percorso della tua directory home. Quindi `cd ~` ti porta sempre in home (come fa `cd` da solo).

TAB Complemento automatico - Quando possibile, Raspbian tenterà di completare il nome del comando che stai scrivendo. Prova a digitare `whoa` seguito da **TAB**. Sarà completato in `whoami`. Funziona anche con i percorsi.

History – hai scordato quel comando? Il tuo Pi ne ha tenuto traccia. Premi il tasto freccia su per ciclare ripetutamente i comandi precedenti. Puoi anche lanciare `history` per avere una lista completa.

!! - Due punti esclamativi significano 'il comando precedente'. Hai dimenticato di usare sudo? non digitare di nuovo il comando, basta inserire sudo `!!`.

47 Usare gli alias

Se c'è un comando lungo che andrai a utilizzare spesso, creagli un alias. Gli alias sono mantenuti in un file nascosto (inesistente per default) chiamato **.bash_aliases**, nella home directory. Per creare o visualizzare i contenuti:

```
nano ~/.bbash_aliases
```

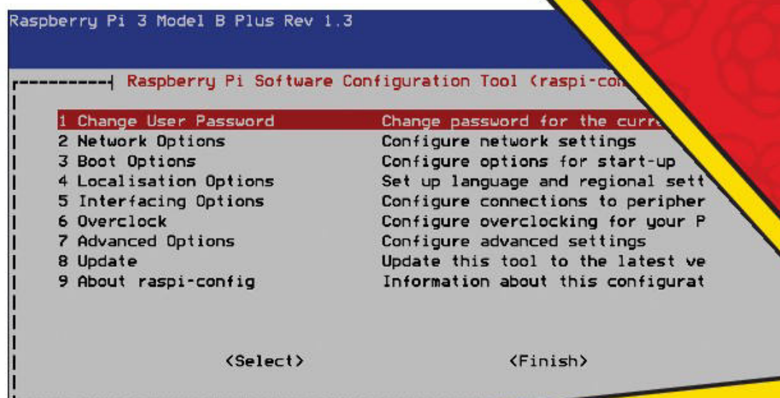
Per ogni alias che vuoi creare, immetti una riga seguendo questo schema:

```
alias howmany="ls -l | wc -l"
```

Ora puoi lanciare `howmany` per questo comando.

48 È bloccato!

Hai un programma congelato sullo schermo? Per accedere nuovamente al prompt, prova **CTRL+C** per interrompere il processo, o **CTRL+Z** per metterlo sullo sfondo. Ora digita `killall <commandname>`. Puoi usare `ps ax` per mostrare un elenco di processi in esecuzione. È ancora lì? Prova `sudo killall <commandname>`. Come ultima spiaggia, usa `sudo kill -9 <processid>` (trova il processid da `ps ax`).



TRUCCHI SULLA RISOLUZIONE DEI PROBLEMI

▲ Qualche applicazione da riga di comando prende un aspetto grafico di base, per essere più facile da usare. La più probabile che tu veda, è raspi-config

49 Scopri di più

Molti comandi sono dotati di una vasta gamma di "argomenti" che ne controllano il comportamento. Sono sotto forma di 'Switch' (semplici flag) o 'parametri' che forniscono informazioni aggiuntive, come un file di input. La maggior parte dei comandi ha una pagina "man" di accompagnamento (abbreviazione di manuale). Per vedere se una 'manpage' esiste per il tuo comando, basta scrivere:

```
man command
```

Puoi anche leggere il manuale di 'man' stesso:

```
man man
```

In alternativa, molti comandi rispondono allo switch `--help` (o `-h`, in breve) fornendo un riepilogo delle opzioni.

50 Entra da remoto

Ci siamo passati tutti: stiamo lavorando felicemente sul Desktop di Raspberry Pi e qualcosa va storto. Ora lo schermo è bloccato, il tuo punteggio record a Doom è perso, e l'unica opzione è scollegare il Pi, che rischia la corruzione dei dati. Bene, potresti essere in grado di entrare usando SSH, il 'Secure SHell' che ti permette di accedere al tuo Pi (a patto che tu abbia già abilitato SSH su di esso) da un altro computer e arrivare alla riga di comando:

```
ssh raspberrypi.local
```

Poi, se arrivi al prompt:

```
sudo shutdown -h now
```

...e il tuo Pi è in salvo.



Parte 03

Programmare un gioco isometrico: AmazeBalls

Pygame Zero in 3D. Creiamo alcuni nemici e della dinamite in questa ultima parte della serie



MAKER

Mark Vanstone

Autore di software educativo degli anni '90, autore della serie ArcVenture, scomparso nella landa desolata del software aziendale. Salvato dal Raspberry Pi!

magpi.cc/YiZnxL

@mindexplorers

Inizieremo da dove eravamo rimasti nell'ultima parte e aggiungeremo alcuni elementi per creare un gioco più impegnativo. Aggiungeremo alcune palle nemiche che vagano nel labirinto, spingendo i muri – quindi anche se sai come arrivare al traguardo, puoi trovare il tuo percorso bloccato. Per dare al nostro giocatore un antidoto al rimanere bloccato, aggiungeremo alcuni candelotti di dinamite da raccogliere e usare.

01 Cambiare i colori

In precedenza avevamo la nostra palla che rimbalzava nel labirinto, spostando la posizione di disegno della mappa in modo tale da vedere sempre l'area intorno alla palla. Ora aggiungeremo altre palle, ma queste lavoreranno in contrapposizione con il giocatore, quindi è necessario colorarle in maniera diversa. Possiamo farlo abbastanza facilmente con un programma di disegno come GIMP. Carica ogni frame e usa lo strumento chiamato 'colorize' (nel menu Colori in GIMP). Assicurati di salvare il frame con un nome differente; per esempio, metti una 'e' davanti al nome del file (e per enemy: "nemico").

02 Riciclare codice

Abbiamo già una palla che rimbalza nel labirinto - per ottenere altre palle, riutilizziamo il codice che abbiamo già. Possiamo duplicare il dizionario dati nella parte iniziale del codice per il giocatore e chiamarlo `enemy1` invece di `player`. Dovrai cambiare i valori `x` e `y` nel file dati in qualcosa come 13, che metterà il nemico circa al centro del labirinto. Ora che abbiamo definito il nostro nemico, possiamo riciclare qualche pezzo di codice.

03 Codice di aggiornamento comune

Possiamo usare, come facciamo per il giocatore, esattamente la stessa funzione `updateBall()`, e questo tratterà tutta l'animazione e il movimento del nemico da un blocco all'altro. Ciò di cui abbiamo bisogno è aggiungere un'altra chiamata a `updateBall()` dopo quella che abbiamo nella funzione Pygame Zero `update()`. Ma questa volta, invece di passare i valori del giocatore alla funzione, passiamo i valori di `enemy1` scrivendo `updateBall(enemy1)`. Questo significa che se impostiamo la palla nemica in movimento, tutte le modifiche ai dati saranno fatte nello stesso modo della palla del giocatore.

04 Disegnare il nemico

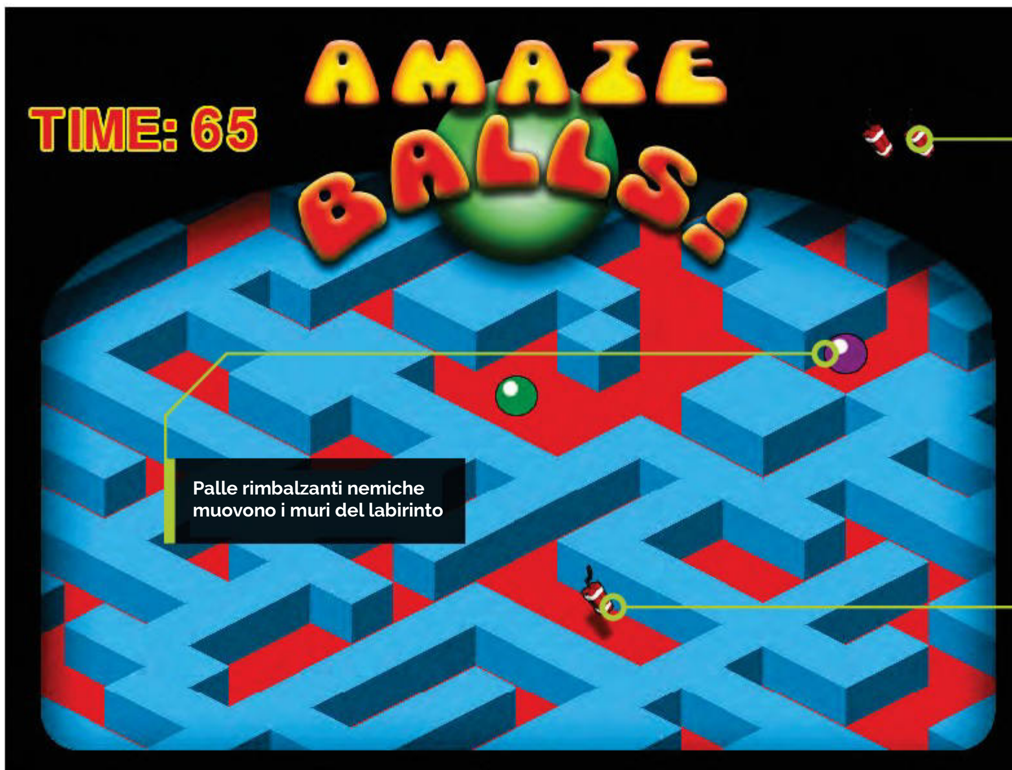
Anche se ora abbiamo un modo per aggiornare l'animazione della palla nemica, dobbiamo anche scrivere del codice per disegnarla sullo schermo. Questo deve essere un po' diverso da quello del giocatore perché la palla nemica deve muoversi mentre la mappa scorre. Dobbiamo usare sia la posizione della mappa che i valori di `sx` e `sy` della palla per capire dove deve essere disegnata. Vedi **figure1.py** per gli aggiornamenti della funzione `drawMap()`. Noterai che viene calcolata la posizione del blocco(`bx` e `by`) e poi aggiunti i valori `sx` e `sy`.

05 Intelligenza dei nemici

Se eseguiamo il nostro programma ora, vedremo una palla nemica che rimbalza in mezzo al labirinto. Noterai che entrambe le palle rimbalzano esattamente allo stesso tempo – se si volesse avere il rimbalzo non sincronizzato, è possibile modificare il valore iniziale del frame

Cosa Serve

- Raspbian Jessie o più recente
- Tiled (editor di mappe gratuito) **mapeditor.org**
- Un programma di manipolazione immagini, come GIMP, o le immagini su **magpi.cc/fPBrhM**
- L' ultima versione di Pygame Zero (1.2)



La dinamite raccolta può essere usata per demolire i muri che bloccano il giocatore

Il giocatore può prendere i candelotti di dinamite

Modifiche a `drawMap()` per incorporare la palla nemica

dichiarato nella parte superiore del codice per `enemy1`. Ora definiamo una funzione chiamata `updateEnemy()`; permetterà alla palla nemica di spostare e spingere alcuni muri intorno a sé.

06 Ottenere movimenti casuali

Dobbiamo far muovere il nemico in modo casuale così, come abbiamo fatto precedentemente in questa serie, usiamo il modulo random per generare un movimento casuale. All'inizio del nostro programma, importiamo il modulo con `from random import randint`. Definiamo la funzione `updateEnemy()` con `def updateEnemy(e):` . La variabile `e` contiene il dizionario dati del nemico che dobbiamo passare alla funzione quando la chiamiamo. Ora definiamo alcune direzioni. Possiamo farlo con una lista di direzioni x e y; per esempio, se avessimo x e y scritto come `[0,1]`, significherebbe muovere di nessun blocco nella direzione x e un blocco nella direzione y.

07 Una direzione

Quindi, possiamo definire tutte e quattro le direzioni come `edirs = [[-1,0], [0,1], [1,0], [0, -1]]`. E poi tutto ciò di cui abbiamo bisogno è sceglierne una con un numero casuale. Per scegliere un numero intero casuale tra 0 e 3, scriviamo `r = randint (0,3)`. Ora possiamo riutilizzare

figure1.py

> Linguaggio: **Python 3**

```
def drawMap():
    psx = OFFSETX
    psy = OFFSETY-32
    mx = psx - player["sx"]
    my = psy - player["sy"]+32

    for x in range(player["x"]-12, player["x"]+16):
        for y in range(player["y"]-12, player["y"]+16):
            if onMap(x,y):
                b = mapData["data"][y][x]
                td = findData(mapData["tiles"], "id", b)
                block = td["image"]
                bheight =  td["imageheight"]-34
                bx = (x*32)-(y*32) + mx
                by = (y*16)+(x*16) + my
                if -32 <= bx < 800 and 100 <= by < 620:
                    screen.blit(block, (bx, by - bheight))
                if x == player["x"] and y == player["y"]:
                    screen.blit("ball"+str(player["frame"]),
(psx, psy))
                if x == enemy1["x"] and y == enemy1["y"]:
                    screen.blit("eball"+str(enemy1[
"frame"]),(bx + enemy1["sx"],(by-32)+enemy1["sy"]))
```

figure2.py

> Linguaggio: **Python 3**

```
def doMove(p, x, y):
    global mazeSolved
    if onMap(p["x"]+x, p["y"]+y):
        mt = mapData["data"][p["y"]+y][p["x"]+x]
        if mt == 1 or mt == 3:
            p.update({"queueX":x, "queueY":y,
"moveDone":False})
            if mt == 3 and p == player:
                mazeSolved = True
        return mt
```

Cambiamenti in `doMove()` per assicurarsi che la palla nemica non inneschi la condizione di fine

Top Tip

Dati dinamici mappa

Si può cambiare qualsiasi blocco sulla mappa cambiando l'`id` in `mapData`. Potresti divertirti molto animando gli elementi della mappa nella funzione `update()`.

la funzione `doMove()` che muove il giocatore, ma usata per il nemico. Abbiamo bisogno di fare due modifiche alla funzione `doMove()`. La prima è per restituire l'`id` del blocco che si sta spostando. Quindi possiamo rilevare se un blocco di muro è sul percorso; se si, possiamo davvero rendere la vita difficile al giocatore, disordinando i blocchi - subdolo, eh?

08 Fare una mossa

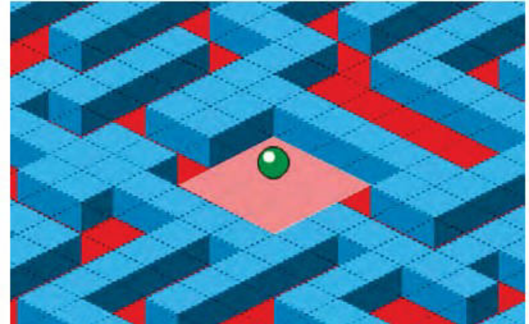
L'altro problema che abbiamo con la funzione `doMove()` è che rileva se il giocatore è arrivato sul traguardo, se usiamo lo stesso codice per il nemico, potremmo ottenere la condizione labirinto risolto innescata dal nemico anziché dal giocatore, quindi dobbiamo cambiare la condizione `mazeSolved` e includere un test per vedere se abbiamo a che fare con il giocatore. Dai un'occhiata a **figure2.py** per vedere queste due modifiche alla funzione `doMove()`. Quando abbiamo apportato queste modifiche, possiamo tornare indietro alla funzione `updateEnemy()`.

09 Blocchi dinamici

Finora, tutti i blocchi sono rimasti nello stesso posto, ma poiché sono tutti rappresentati come numeri nella mappa dati, possiamo cambiare i numeri nei dati e vedremo il cambiamento del labirinto sullo schermo. Ad esempio, se un blocco ha `id` 2, è un muro di mattoni. Se cambiamo il dato in `id` 1, vedremo sparire il muro e ci sarà un blocco pavimento al suo posto. Quindi, possiamo apportare modifiche al labirinto mentre la palla nemica rimbalza.



Ci sono diversi modi di creare immagini per i giochi. La dinamite è stata creata con un programma di modellazione 3D gratuito chiamato Blender



Quando viene utilizzata la dinamite, essa modifica tutti i blocchi intorno al giocatore (mostrati qui in rosa) in blocchi pavimento

10 Spostamento dei muri

Quindi, tornando alla nostra funzione `updateEnemy()`, dobbiamo innanzitutto generare un numero casuale per la direzione e quindi spostare la palla nemica in quella direzione. Ma se si muove verso un muro, dobbiamo spostare quel muro nella direzione in cui si muove la palla. L'altra cosa che dobbiamo controllare è se c'è uno spazio dove il muro deve spostarsi. Dobbiamo chiamare la funzione `doMove()` usando il dizionario di `enemy1` (passato in `updateEnemy()`) e quindi, se l'`id` del blocco è 2 (un muro), chiamare un'altra funzione chiamata `moveBlock()`.

11 Cambiare posto

Dobbiamo definire la funzione `moveBlock()` alla quale passeremo i dati del blocco x e y nella mappa e anche il valore della direzione che stiamo usando per spostare la palla nemica. Innanzitutto, la funzione controllerà che ci stiamo muovendo all'interno dell'area della mappa e quindi controllerà che il blocco su cui stiamo spostando il muro è un pavimento (`id` 1). Se tutto questo si verifica, allora copiamo l'`id` del blocco nel quale ci stiamo spostando nella nuova posizione. Vedi la **figure3.py** per vedere la funzione `updateEnemy()` e la funzione `moveBlock()`.

12 Cosa stiamo facendo?

Il codice in **figure3.py** può sembrare un po' scoraggiante in alcuni punti, quindi diamo uno sguardo ai dettagli. La funzione `updateEnemy()` sta fondamentalmente definendo quattro direzioni per muoversi e poi diciamo: se il nemico non si sta attualmente muovendo, allora prendi una direzione casuale e passa i valori di x e y di questa direzione alla funzione `doMove()`. Se il blocco dove ci stiamo muovendo ha `id` 2, allora muovi il blocco usando la posizione in cui ci stiamo spostando e il valore della direzione. Quindi azzera le coordinate dello schermo x e y (`sx` e `sy`) nel dizionario del nemico.

13 E' tutto relativo

Noterai che se il nemico si sta muovendo, allora controlliamo se siamo al frame sette (quando la palla si muove effettivamente da un blocco ad un altro nei dati) e se è così, sistemiamo le coordinate in modo che siano ora relative alla nuova posizione della mappa piuttosto che alla vecchia. La funzione **`moveBlock()`** controlla direttamente i dati in **`mapData`** per verificare che il blocco possa essere spostato, sposta i dati dalla posizione di origine alla posizione di destinazione e trasforma la posizione di origine in un blocco di pavimento.

14 Nemici multipli!

Una volta fatto tutto, dobbiamo solo aggiungere **`updateEnemy(enemy1)`** dopo le chiamate a **`updateBall()`** nella funzione **`update()`** di Pygame Zero. Ora, può essere che consideriamo una sola palla nemica non sufficiente per rendere il gioco interessante e crearne una seconda è adesso molto semplice. Dobbiamo solo duplicare il dizionario **`enemy1`** e chiamarlo **`enemy2`**, cambiare **`x`** e **`y`** iniziali (forse a 25), effettuare chiamate a **`updateBall(enemy2)`** e **`updateEnemy(enemy2)`** nella funzione **`update()`** e prima che tu te ne renda conto, hai una seconda palla nemica. Puoi farne quante ne vuoi, o magari metterle in una lista per maggiore efficienza se ce ne sono più di tre.

15 Un po' impari

Ora che abbiamo i cattivi a scombinare il labirinto, diventerà piuttosto difficile per il nostro giocatore arrivare al traguardo, quindi è tempo di livellare il campo di gioco, in questo caso letteralmente. Introduciamo un po' di dinamite nel mix! Bene, dobbiamo fare un blocco grafico per la dinamite da poter usare nell'editor di mappe Tiled e anche una icona che possiamo usare per mostrare quanti candelotti di dinamite il giocatore ha raccolto. Se desideri utilizzare grafica e dati già pronti, sono disponibili nel repository GitHub: **magpi.cc/NvafjA**.

16 Maneggiare gli esplosivi

Innanzitutto, aggiungiamo una variabile per contare il numero di candelotti di dinamite in possesso del giocatore, può essere fatto scrivendo **`"dynamite": 0`** come parte del dizionario **`player`**. Quindi, assumendo di aver aggiunto della dinamite

figure3.py

> Linguaggio: **Python 3**

```python
def updateEnemy(e):
    edirs = [[-1,0],[0,1],[1,0],[0,-1]]
    if e["moveX"] == 0 and e["moveY"] == 0:
        r = randint(0,3)
        if doMove(e, edirs[r][0], edirs[r][1]) == 2:
            moveBlock(e["x"]+edirs[r][0],e["y"]+edirs[r][1],edirs[r][0],edirs[r][1])
        e["sx"] = e["sy"] = 0
    else:
        if e["frame"] == 7 and e["movingNow"] == True:
            if e["sx"] == 12: e["sx"] -= 32
            if e["sx"] == -12: e["sx"] += 32
            if e["sy"] == 6: e["sy"] -= 16
            if e["sy"] == -6: e["sy"] += 16

def moveBlock(mx,my,dx,dy):
    if onMap(mx+dx,my+dy):
        d = mapData["data"][my+dy][mx+dx]
        if d == 1:
            mapData["data"][my+dy][mx+dx] = mapData["data"][my][mx]
            mapData["data"][my][mx] = 1
```

Aggiornare la palla nemica e spostare i blocchi se i muri sono di ostacolo

La funzione **`update()`** aggiornata per includere due palle nemiche

figure4.py

> Linguaggio: **Python 3**

```python
def update(): # Funzione aggiornamento Pygame Zero
    global player, timer
    mt = 0
    if player["moveDone"] == True:
        if keyboard.left: mt = doMove(player, -1, 0)
        if keyboard.right: mt = doMove(player, 1, 0)
        if keyboard.up: mt = doMove(player, 0, -1)
        if keyboard.down: mt = doMove(player, 0, 1)
    if mt == 4:
        mapData["data"][ player["y"] + player["queueY"]][ player["x"] + player["queueX"]] = 1
        player["dynamite"] += 1
    updateBall(player)
    updateBall(enemy1)
    updateBall(enemy2)
    updateEnemy(enemy1)
    updateEnemy(enemy2)
```

ai nostri dati della mappa (vedi la parte precedente di questa serie per i dettagli sulla modifica con Tiled), dobbiamo rilevare se il nostro giocatore si è spostato su un blocco di dinamite e, in tal caso, aggiungere `1` al nostro contatore `dynamite` e sostituire il blocco dinamite con un blocco di pavimento, che lo farà scomparire dalla mappa. Possiamo farlo nella funzione `update()`.

17 Scorta di munizioni

Per gestire la raccolta, abbiamo solo bisogno di testare il valore di `mt` dopo i controlli della tastiera. Vedere **figure4.py** per visualizzare la funzione `update()` aggiornata. Quando il nostro giocatore ha preso alcuni candelotti di dinamite, possiamo mostrarne il numero con delle icone, come abbiamo fatto prima (per esempio con le vite), nella funzione `draw()` scrivendo `for l in range(player["dynamite"]): screen.blit("dmicon", (650+(l*32),80))` che disegnerà le icone della dinamite in alto a destra allo schermo. Quindi ora abbiamo le munizioni che il giocatore può usare per aprire un percorso tra i blocchi che i nemici hanno messo in mezzo.

18 Andando via con il botto

Tutto quello che ci resta da fare ora è codificare il meccanismo per scatenare la dinamite. Lo facciamo con la funzione Pygame Zero `on_key_down()`. Dobbiamo testare se la barra dello **SPAZIO** è stata premuta e, in tal caso, liberare uno spazio intorno al giocatore, sostituendo tutti i blocchi con blocchi di pavimento. Può essere fatto con un ciclo `for` annidato. Dai un'occhiata al listato completo di **amazeballs3.py** per vedere questo ultimo pezzo di codice.

Ora è il momento di testare come è impostato il gioco, è troppo facile o troppo difficile? Hai bisogno di più o meno nemici? Potresti provare a fare una serie di mappe differenti con altri oggetti da collezionare.

19 Ed infine

Bene, purtroppo è tutto per il momento, per questa serie. Speriamo tu abbia imparato molto su Pygame Zero e sui giochi scritti in Python. Noi dobbiamo ringraziare con entusiasmo il creatore di Pygame Zero, Daniel Pope: senza il suo lavoro eccellente, questa serie non sarebbe esistita. Speriamo che tu concorda con noi che il framework Pygame Zero è un punto di partenza ideale per imparare a programmare giochi su Raspberry Pi.

amazeballs3.py

> Linguaggio: **Python**

```python
import pgzrun
import map3d
from random import randint

player = {"x":3, "y":3, "frame":0, "sx":0, "sy":96,
          "moveX":0, "moveY":0, "queueX":0, "queueY":0,
          "moveDone":True, "movingNow":False, "animCounter":0, "dynamite":0}
enemy1 = {"x":13, "y":13, "frame":0, "sx":0, "sy":0,
          "moveX":0, "moveY":0, "queueX":0, "queueY":0,
          "moveDone":True, "movingNow":False, "animCounter":0}
enemy2 = {"x":25, "y":25, "frame":0, "sx":0, "sy":0,
          "moveX":0, "moveY":0, "queueX":0, "queueY":0,
          "moveDone":True, "movingNow":False, "animCounter":0}
OFFSETX = 368
OFFSETY = 300
timer = 0
mazeSolved = False

mapData = map3d.loadmap("maps/map1.json")

def draw(): # Funzione di disegno Pygame Zero
    screen.fill((0, 0, 0))
    drawMap()
    screen.blit('title', (0, 0))
    screen.draw.text("TIME: "+str(timer) , topleft=(20, 80), owidth=0.5, ocolor=(255,255,0), color=(255,0,0) , fontsize=60)
    for l in range(player["dynamite"]): screen.blit("dmicon", (650+(l*32),80))
    if mazeSolved:
        screen.draw.text("MAZE SOLVED in " + str(timer) + " seconds!" , center=(400, 450), owidth=0.5, ocolor=(0,0,0), color=(0,255,0) , fontsize=60)


def update(): # Funzione di aggiornamento Pygame Zero
    global player, timer
    mt = 0
    if player["moveDone"] == True:
        if keyboard.left: mt = doMove(player, -1, 0)
        if keyboard.right: mt = doMove(player, 1, 0)
        if keyboard.up: mt = doMove(player, 0, -1)
        if keyboard.down: mt = doMove(player, 0, 1)
    if mt == 4:
        mapData["data"][ player["y"] + player["queueY"]][player["x"] + player["queueX"]] = 1
        player["dynamite"] += 1
    updateBall(player)
    updateBall(enemy1)
    updateBall(enemy2)
    updateEnemy(enemy1)
    updateEnemy(enemy2)

def on_key_down(key):
    if player["dynamite"] > 0 and key.name == "SPACE":
        player["dynamite"] -= 1
        for x in range(player["x"]-1, player["x"]+2):
```

SCARICA IL CODICE COMPLETO:

 **magpi.cc/ODgUCg**

```
            for y in range(player["y"]-1, player["y"]+2):
                mapData["data"][y][x] = 1

def timerTick():
    global timer
    if not mazeSolved:
        timer += 1

def drawMap():
    psx = OFFSETX
    psy = OFFSETY-32
    mx = psx - player["sx"]
    my = psy - player["sy"]+32

    for x in range(player["x"]-12, player["x"]+16):
        for y in range(player["y"]-12, player["y"]+16):
            if onMap(x,y):
                b = mapData["data"][y][x]
                td = findData(mapData["tiles"], "id", b)
                block = td["image"]
                bheight =  td["imageheight"]-34
                bx = (x*32)-(y*32) + mx
                by = (y*16)+(x*16) + my
                if -32 <= bx < 800 and 100 <= by < 620:
                    screen.blit(block, (bx, by - bheight))
                if x == player["x"] and y == player["y"]:
                    screen.blit("ball"+str(player["frame"]), (psx, psy))
                if x == enemy1["x"] and y == enemy1["y"]:
                    screen.blit("eball"+str(enemy1["frame"]), (bx + enemy1["sx"], (by-32)+enemy1["sy"]))
                if x == enemy2["x"] and y == enemy2["y"]:
                    screen.blit("eball"+str(enemy2["frame"]), (bx + enemy2["sx"], (by-32)+enemy2["sy"]))

def findData(lst, key, value):
    for i, dic in enumerate(lst):
        if dic[key] == value:
            return dic
    return -1

def onMap(x,y):
    if 0 <= x < mapData["width"] and 0 <= y < mapData["height"]:
        return True
    return False

def doMove(p, x, y):
    global mazeSolved
    if onMap(p["x"]+x, p["y"]+y):
        mt = mapData["data"][p["y"]+y][p["x"]+x]
        if mt == 1 or mt == 3 or mt == 4:
            p.update({"queueX":x, "queueY":y, "moveDone":False})
            if mt == 3 and p == player:
                mazeSolved = True
        return mt

def updateEnemy(e):
    edirs = [[-1,0],[0,1],[1,0],[0,-1]]
    if e["moveX"] == 0 and e["moveY"] == 0:
        r = randint(0,3)
        if doMove(e, edirs[r][0], edirs[r][1]) == 2:
            moveBlock(e["x"]+edirs[r][0],e["y"]+edirs[r][1],edirs[r][0],edirs[r][1])
        e["sx"] = e["sy"] = 0
    else:
        if e["frame"] == 7 and e["movingNow"] == True:
            if e["sx"] == 12: e["sx"] -= 32
            if e["sx"] == -12: e["sx"] += 32
            if e["sy"] == 6: e["sy"] -= 16
            if e["sy"] == -6: e["sy"] += 16

def moveBlock(mx,my,dx,dy):
    if onMap(mx+dx,my+dy):
        d = mapData["data"][my+dy][mx+dx]
        if d == 1:
            mapData["data"][my+dy][mx+dx] = mapData["data"][my][mx]
            mapData["data"][my][mx] = 1

def updateBall(p):
    if p["movingNow"]:
        if p["moveX"] == -1: moveP(p,-1,-0.5)
        if p["moveX"] == 1: moveP(p,1,0.5)
        if p["moveY"] == -1: moveP(p,1,-0.5)
        if p["moveY"] == 1: moveP(p,-1,0.5)
    p["animCounter"] += 1
    if p["animCounter"] == 4:
        p["animCounter"] = 0
        p["frame"] += 1
        if p["frame"] > 7:
            p["frame"] = 0
        if p["frame"] == 4:
            if p["moveDone"] == False:
                if p["queueX"] != 0 or p["queueY"] !=0:
                    p.update({"moveX":p["queueX"], "moveY":p["queueY"], "queueX":0, "queueY":0, "movingNow": True})
            else:
                p.update({"moveDone":True, "moveX":0, "moveY":0, "movingNow":False})

        if p["frame"] == 7 and p["moveDone"] == False and p["movingNow"] == True:
            p["x"] += p["moveX"]
            p["y"] += p["moveY"]
            p["moveDone"] = True

def moveP(p,x,y):
    p["sx"] += x
    p["sy"] += y

clock.schedule_interval(timerTick, 1.0)
pgzrun.go()
```

FANTASTICI PROGETTI HAM RADIO

Raspberry Pi e radioamatori – un incontro nel paradiso degli hobbisti. Di **Rob Zwetsloot**

Se sei un lettore regolare di The MagPi, avrai notato un sondaggio nella tua casella di posta elettronica un po' di tempo fa, riguardo la rivista. Una delle domande riguardava il tipo di argomenti che ti piacerebbe trovare nella rivista, e abbiamo avuto un numero travolgente di richieste di trattare l'argomento radio amatoriali.

Da quando è uscito il Pi, la comunità radioamatoriale lo ha usato per divertimento e talvolta per progetti pratici, quindi è giunto il momento di farti sapere quali cose divertenti può fare il tuo Pi con il mondo dei radioamatori.

COSA È LA HAM RADIO?

La radio amatoriale ha una lunga storia. Cosa lo rende diverso dalla normale radio?

Da quando la radio è stata utilizzata per comunicare, ci sono stati dilettanti appassionati di radio. Così come la radio è diventata sempre più diffusa nei primi decenni del 20° secolo, altrettanto ha fatto l'uso della radio amatoriale. Con il crescere del numero di appassionati, sono cresciute anche le possibilità di utenti radioamatori maliziosi, che, alla fine, hanno portato a leggi in tutto il mondo atte a limitare la frequenza con cui queste "ham radio" possono operare.

Nel corso degli anni, le frequenze sono cambiate, e sono state istituire nuove regole di registrazione e licenza per assicurarsi che la comunità radio amatoriale sappia bene come divertirsi col proprio hobby senza interrompere nulla di importante.

Era moderna

Con il recente boom di dispositivi smart, micro computer e altri dispositivi elettronici hobbistici, le possibilità e l'accessibilità del radioamatore sono solo aumentate. Grazie a componenti più economiche e più potenza con cui divertirsi, abbiamo visto molti progetti Pi, nel corso degli anni, che fanno uso delle ham radio. Speriamo di poterti mostrare qualcosa che non hai mai visto prima nelle prossime pagine, o darti l'ispirazione per il tuo progetto da radioamatore.

Perché ham?

Nei primi giorni dei telegrafi, chi non nera così abile con il codice Morse erano spesso indicato come "ham" (prosciutto"), alla stregua di un insulto. Molti modi di dire contenevano "ham" in termine peggiorativo (in italiano ad esempio "orecchie foderate di prosciutto), è stato poi reclamato dalla comunità radio amatoriale - e questo è come si è arrivati a ham radio.



▲ Molto personale dell'esercito doveva usa le ham radio per comunicare con i propri cari durante la metà del 20° secolo



▲ I tecnici degli uffici postali ispezionano le apparecchiature radio nel 1897. ***Crediti: Cardiff Flat Holm Project***



◀ L'appassionato radioamatore tedesco affronta le piste con la sua attrezzatura 'portatile'

USARE HAM RADIO CON RASPBERRY PI

Cosa ti serve per partire a utilizzare applicazioni da radioamatore

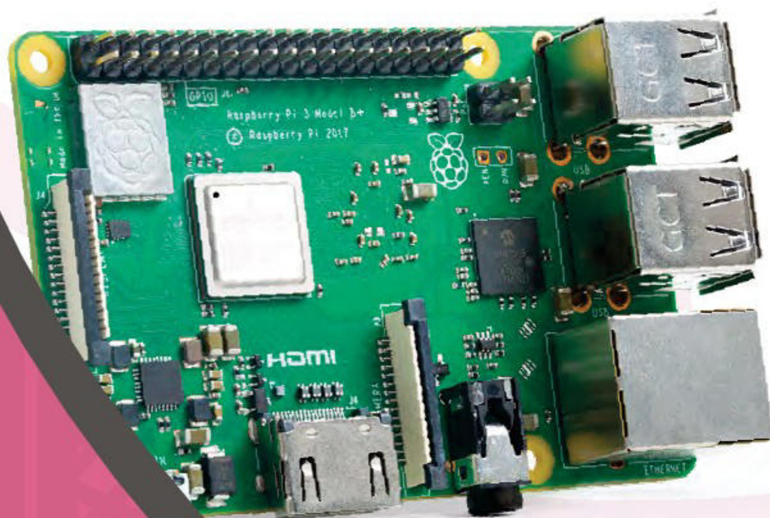
Visto che Raspberry Pi è stato così popolare con i radioamatori per così tanto tempo, ci sono un sacco di software e di risorse prontamente disponibili per le persone che vogliono dilettarsi con la radio gestita dal Pi. Poiché Raspbian è basato su Linux, significa anche che molti pacchetti esistenti sono stati portati sul Pi.

Cosa ti serve...

UN QUALSIASI RASPBERRY PI – Più recente è, meglio è, ma molte applicazioni funzioneranno su un Pi Zero così come su Raspberry Pi 3B+.

RASPBIAN – La maggior parte del software importante sarà compatibile con Raspbian, quindi se non lo stai usando, è una buona idea preparare una scheda SD con esso.

LICENZA DA RADIOAMATORE – È contro la legge nella maggior parte dei paesi operare come radioamatore senza le relative licenze di trasmissione.



Un Raspberry Pi 3 Modello B+ è consigliato, Anche se un Pi Zero può gestire comunque molte applicazioni radio

ATTENZIONE!

Hai bisogno di una licenza per operare come radioamatore nella maggior parte dei paesi. Ci sono molte leggi e regolamenti che è necessario seguire, altrimenti potresti causare problemi per gli altri. Dai una occhiata su come ottenere una licenza in Italia, vai al sito del Ministero dello sviluppo economico qui: **bit.ly/2VZ5v3Q**

Uso di base

Le onde radio sono uno dei primi metodi usati per codificare dei segnali. Così come la fibra ottica usa schemi luminosi per trasmettere i dati e il filo di rame usa impulsi elettrici, le onde radio sono modulate in modi specifici per contenere i dati. La forma più comune è l'audio e specifiche tipologie di modulazione (ampiezza e frequenza) sono migliori a seconda dello scopo.

Nel contesto dell'uso della radio amatoriale con Raspberry Pi, lo puoi usare per codificare e decodificare le onde radio nello stesso modo in cui farebbe dell'hardware dedicato. L'utilizzo di un computer completo per questo compito, consente maggior flessibilità nel modo di leggere e inviare dati - per esempio, con la radio nella tua auto non potresti essere in grado di visualizzare le rotte degli aerei nel cielo sopra di te.

Vantaggi Ham Pi

Il Raspberry Pi è un ottimo computer da usare in congiunzione con la radio amatoriale grazie alla sua hackerabilità, al basso consumo, e alla portabilità. Puoi facilmente leggere e persino emettere dati rilevanti tramite i pin del GPIO con il corretto software, e portare in giro dei dispositivi che saranno completamente funzionali, che dureranno un bel po' di tempo con batteria portatile o similare.

Nelle prossime pagine, ti mostreremo alcune delle cose che puoi fare con un Raspberry Pi e le ham radio.

FANTASTICI PROGETTI HAM RADIO

Non sai bene dove cominciare? Dai uno sguardo a queste guide!

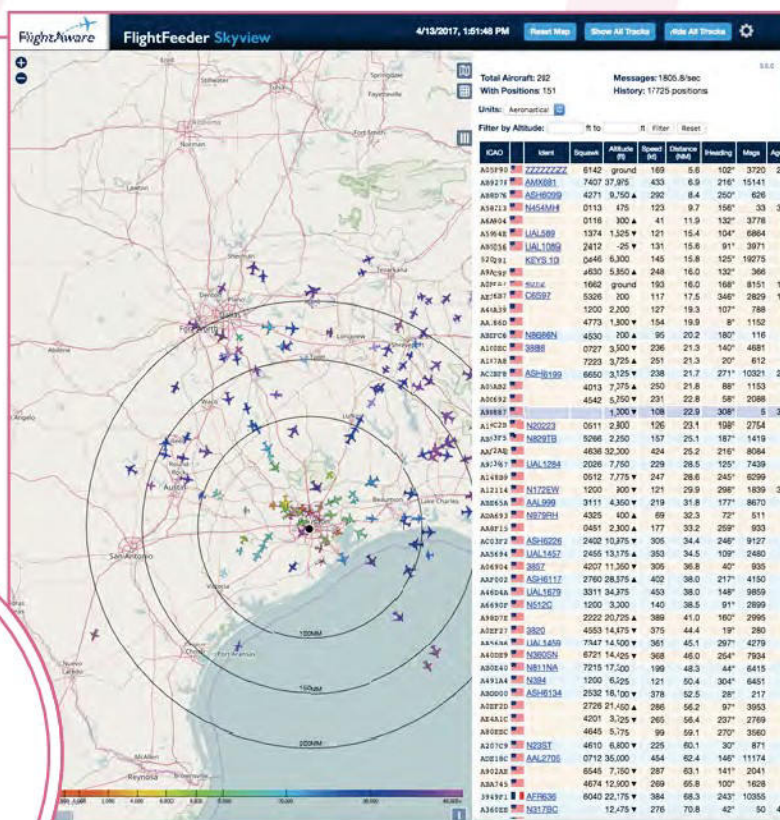
ADS-B flight tracker

Questo è uno dei progetti radioamatoriali più comuni che trovi per Raspberry Pi: trasformando il Pi in una stazione di terra ADS-B, ti consente di tracciare i voli. Una tecnologia simile è utilizzata nei progetti di ballooning ad alta quota per tracciare la posizione del pallone.

Lavora in campo visivo e può avere una portata fino a 300 miglia. Utilizzando i dati radio e Il software PiAware di FlightAware, è quindi possibile tracciare gli aerei nella tua zona.

magpi.cc/phwYXr

> È utile per vedere se l'aereo di un amico o di qualcuno che ami è in ritardo



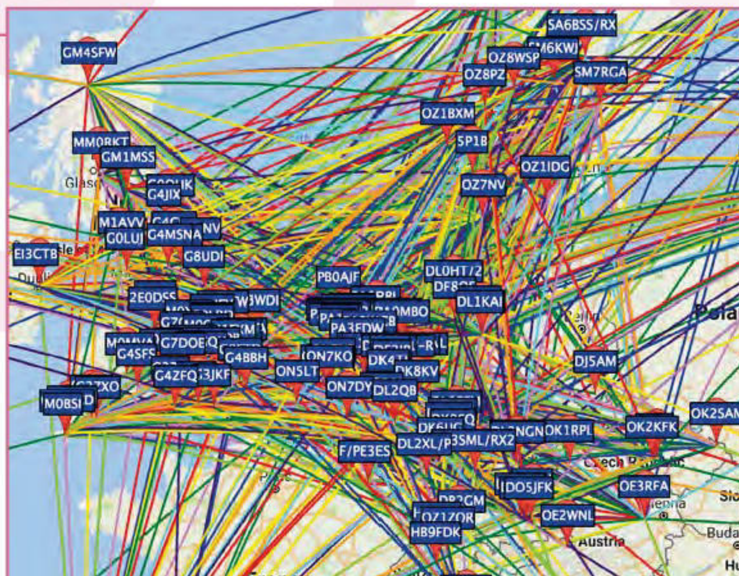
▲ Usa PiAware per trasformare il tuo Raspberry Pi in una stazione di terra ADS-B per tracciare i voli aerei in un raggio di 300 miglia

Trasmettitore WSPR

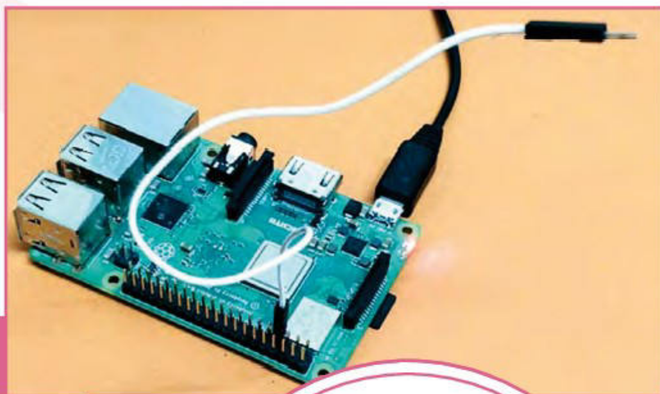
I beacon WSPR (o ‘whisper’) sono abbastanza comuni su Raspberry Pi, permettendo agli appassionati di ham radio di creare una mappa di rete di dove sono le persone e la gamma dei loro dispositivi con frequenze specifiche, così da poter capire quanto siano utili quelle frequenze. Puoi trovare la mappa e altri dettagli su **magpi.cc/uJnbUN**.

WSPR sta per "weak signal propagation reporter" (rapporto propagazione segnale debole), e fa uso di trasmissioni a bassa potenza per creare il suo rapporto. C'è un software che lo usa, fatto appositamente per Raspberry Pi, chiamato WsprryPi, e funziona un po' come un trasmettitore FM, richiede solo una antenna per svolgere la sua funzione.

magpi.cc/EjQpbd



Collega un paio di pin GPIO del Pi, attraverso un filtro passa-basso, ad un'antenna e anche tu puoi essere sulla mappa della rete WSPR



Trasmettitore FM

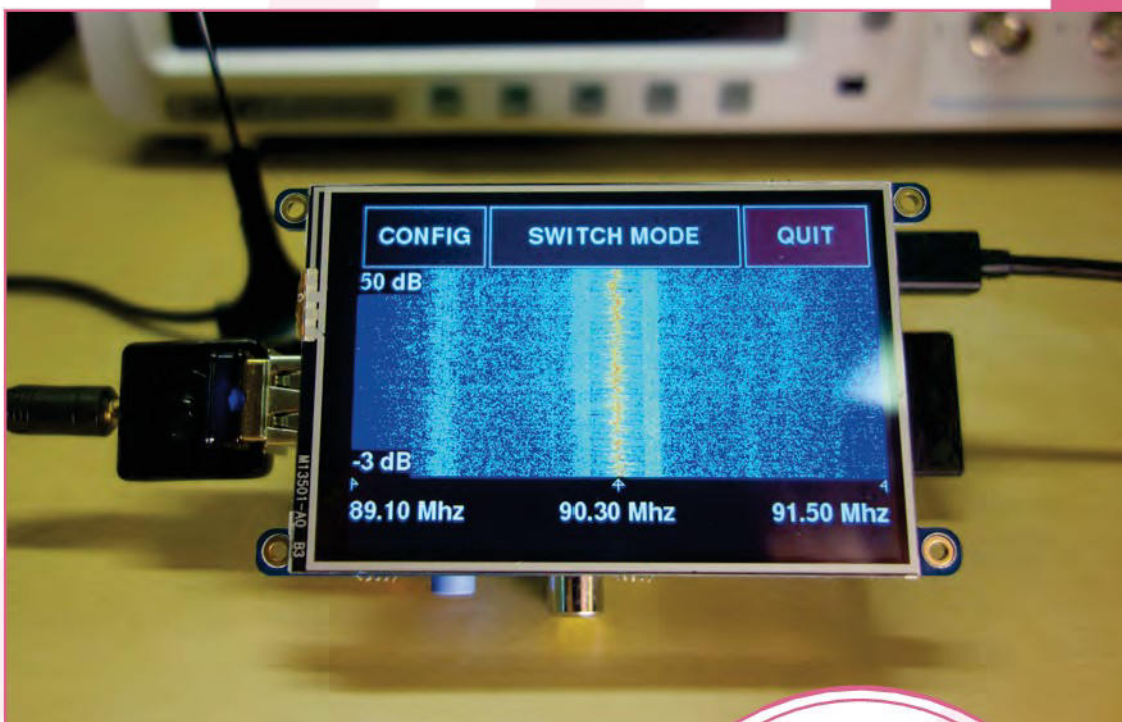
Un divertente progetto che ti permette di creare la tua stazione radio personale o una specie di walkie-talkie - anche se con un raggio estremamente limitato (parliamo di metri). Alcuni dicono che è un'ottima soluzione per trasmettere l’audio dei film nei drive-in, permettendoti di sintonizzare il film con la tua auto, come se fossi negli anni Cinquanta.

Funziona usando il segnale di clock del Pi, che fa parte del processore, per creare un segnale FM che può essere trasmesso a una breve distanza. Dal punto di vista dell'hardware, tutto quel che serve, è un'antenna collegata a uno specifico pin GPIO - in questo caso, un cavo jumper.

magpi.cc/RVSife

> “Un divertente progetto che ti permette di crea la tua personale stazione radio”

Non dimenticare di guardare anche come ricevere immagini dallo spazio usando ham radio a pagina 62! (non compreso in questo estratto in italiano)



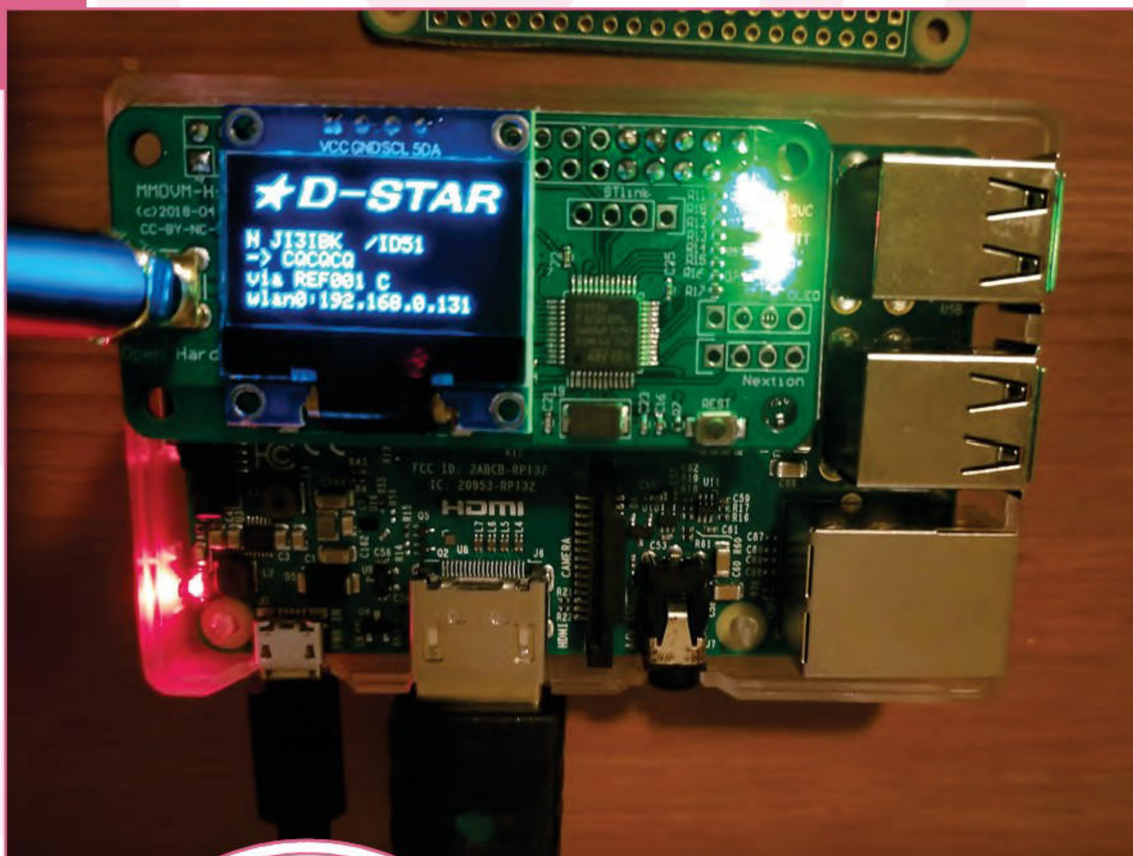
Scanner SDR remoto

La radio definita dal software, o SDR, è dove il il computer decide come decodificare un segnale radio, piuttosto che qualsiasi specifico hardware radio collegato ad esso. Questo aggiunge uno strato di flessibilità alla ricezione di un segnale, ed è qualcosa che puoi creare con un Raspberry Pi.

Sebbene affidabile dal punto di vista del software, richiede l'uso di hardware specifico per il ricevitore, sotto forma di un dongle USB RTL-SDR per ottenere effettivamente i segnali da elaborare. SDR# (leggi SDR sharp) è un popolare software per questo e la guida contiene come configurarlo su Pi.

magpi.cc/jawZRE

> L' SDR è dove il computer decide come decodificare un segnale radio



> "È in grado di inoltrare trasmissioni digitali attraverso la rete"

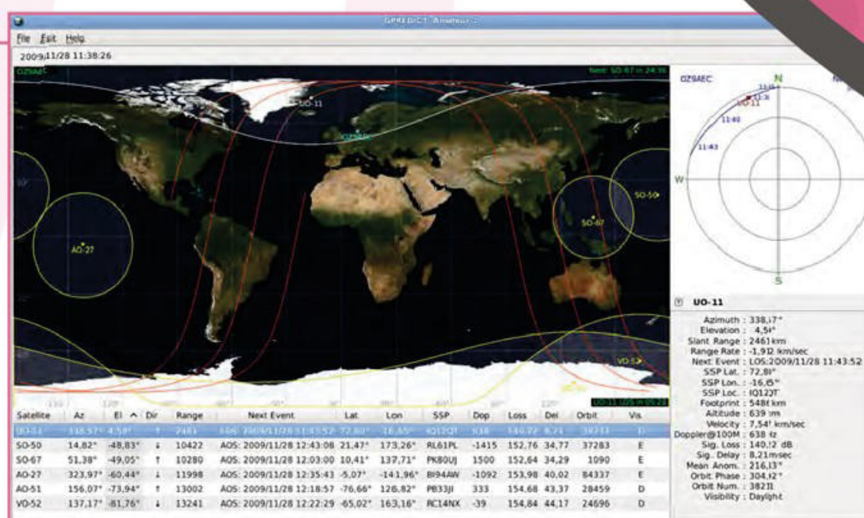
Hotspot voice digitale

La voce digitale consente la codifica e la decodifica di trasmissioni vocali radioamatoriali. Più o meno allo stesso modo in cui la televisione è passata dall'analogico al digitale, questo ha i vantaggi sul miglioramento del segnale, mentre utilizza, allo stesso tempo, anche una minore larghezza di banda. Ci sono alcune specifiche per questo tipo di utilizzo, e questo progetto si avvale di D-STAR per gestirlo.

Avrai ancora bisogno di hardware specifico per questo, e il cavo jumper come antenna non funzionerà in questo caso. Oltre a lavorare per le tue trasmissioni, è in grado di inoltrare trasmissioni digitali attraverso la rete.

magpi.cc/VjYZcp

Hai realizzato un tuo meraviglioso progetto ham radio? Twittalo a @TheMagPi

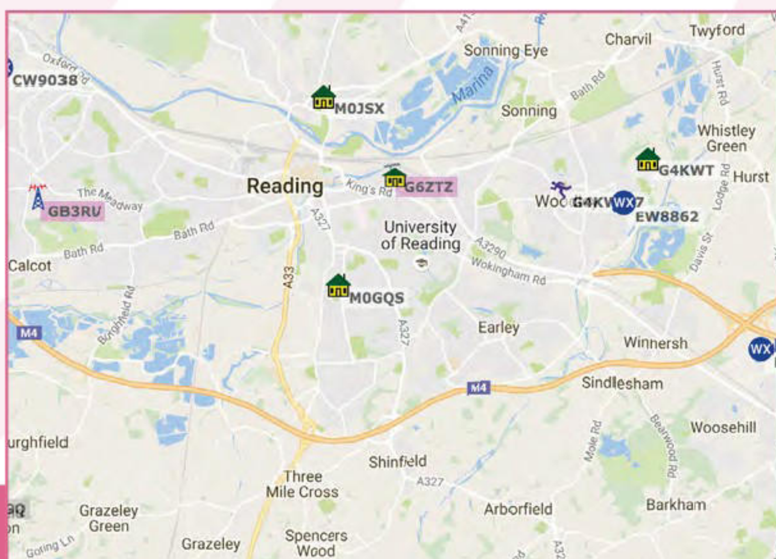


Tracciamento satelliti

Pensa al tracker aereo ma su scala molto più grande - utilizzando il software Gpredict, è possibile tenere traccia dei satelliti. Nello spazio! È troppo figo. Include anche un ottimo software di visualizzazione che è altamente personalizzabile. Abbiamo visto, in passato, usare questo tipo di software per tracciare la ISS - un buon modo per essere sicuro di trovarti pronto e all'esterno se passa in sorvolo durante la notte!

È anche abbastanza potente da controllare le antenne, se hai quelle mobili collegate. Avrai bisogno però di compilalo su Raspberry Pi, quindi è un progetto molto più avanzato.

magpi.cc/BqPWLm



Con un Raspberry Pi e un dongle RTL-SDR, puoi configurare il tuo APRS IGate personale

APRS IGate

Per loro stessa natura, alcune apparecchiature radioamatoriali non sono particolarmente potenti. Inoltre, le onde radio degradano a causa di interferenze e fisica. Quello che un Automatic Packet Reporting System (APRS) fa, è permettere a pacchetti da altri operatori radio di essere digitalmente ripetuti online (Internet Gateway o IGate), possono quindi viaggiare molto più lontano rispetto all'essere trasmessi da qualche altra parte. È anche un buon modo per rintracciare te stesso e altri radioamatori locali.

Ancora una volta, la configurazione è incredibilmente simile a quella di SDR, usando lo stesso dongle e software, anche se con alcune modifiche al modo in cui i dati vengono gestiti.

magpi.cc/yppPRB